



📞 302.335.8868

✉️ [admissions@brightstarinstitute.com](mailto:admissions@brightstarinstitute.com)

## Introduction to Programming Concepts

**Duration:** 15 hours

**Prerequisites:** Rudimentary knowledge of computer systems. Students should know how to basically operate a computer, browse the world wide web, send and receive email messages, and write and print memos.

**Description:** Courses that teach programming languages tend to focus on the particulars of each language (as they should), expending only minimal time discussing basic universal programming concepts. Students who are new to programming may find themselves overwhelmed in discussions of the details of matters such as control flow operators, executable images, third generation languages, and logical operators before they really understand the concepts. This situation can leave the student lost or frustrated or both.

This course is offered for the aspiring programmer to provide them the foundational knowledge that is necessary to learn and advance in a programming career. Students will be exposed to fundamental concepts that are utilized by all programming languages in this course.

### Course Overview

#### The Essence of Programming

- The language of computers
- Computer components
- Hardware and software
- Operating systems
- Programs

#### Types of Programming

- Computer languages

#### The Tasks of Programmers

- Analysis and requirements gathering
- Design and planning
- Coding
- Testing
- Deployment/Documentation
- Maintenance
- Your job

#### Statements and Storage

- The organization of programs

- Machine code
- Early programming
- The evolution of programming languages
- First Generation Languages
- Second Generation Languages
- Third Generation Languages
- Fourth Generation Languages
- Fifth Generation Languages
- Compilers
- The process of compiling
- Practical compiling
- Interpreted languages
- Pros and cons of compiled and interpreted languages
- Imperative programming
- Structured programming
- Procedural programming
- Object-oriented programming
- Declarative programming
- Functional programming
- Stand-alone programs
- Distributed programming
- Client-server programming
- Web-enabled programming
- Textual programming
- IDEs
- Visual programming
- Libraries
- Declaring and initializing variables and constants
- Data types: integers, floats, Booleans, characters, arrays, structures, objects, pointers
- Proper data typing
- Allocation
- Static and dynamic data typing
- Strong and weak data typing
- Scope
- Inheritance

## Operators and Control Flow

- Assignment operators
- Arithmetic operators
- Relational operators
- Conditional operators
- Logical operators
- Bitwise operators
- Special operators
- *if*, *if-then*, *if-then-else* statements
- *case* or *switch-case* statements
- *for* or *foreach* statements
- *while* statements

## The Language of Programming

- Algorithms
- Strings
- Regular expressions
- Input/Output
- Graphical user interfaces
- Events
- Frameworks
- Debugging
- Compile time vs. run time errors
- Database programming
- Technical communities

*do-while* statements

Benchmarks

- Protocols

## The Real World

- Working with the *Command Prompt*
- Customizing your environment
- Help
- Online documentation
- Batch files and scripts
- Installing and using a compiler
- Man pages
- Writing a C program
- Debugging C programs

---

302.335.8868

[admissions@brightstarinstitute.com](mailto:admissions@brightstarinstitute.com)

Copyright© Bright Star Institute